ITIL®

Problem Management
ITIL®4 Practice Guide

AXELOS.com

AXELOS
GLOBAL BEST PRACTICE

22 November 2019

Contents

1 About this document

This document provides practical guidance for the problem management practice. It is split into five main sections, covering:

- general information about the practice
- the practice's processes and activities and their roles in the service value chain
- the organizations and people involved in the practice
- the information and technology supporting the practice
- considerations for partners and suppliers for the practice.

1.1 ITIL® 4 QUALIFICATION SCHEME

Selected content from this document is examinable as a part of the following syllabus:

- ITIL Specialist Create, Deliver and Support

Please refer to the syllabus document for details.

2 General information

2.1 PURPOSE AND DESCRIPTION

Key message

The purpose of the problem management practice is to reduce the likelihood and impact of incidents by identifying actual and potential causes of incidents, and managing workarounds and known errors.

No service is perfect. Every service has errors or flaws which can cause incidents. Errors may originate in any of the four dimensions of service management. For example, a mistake in a third-party contract is as likely to cause an incident as a component failure. Many errors are identified before a service goes live and are then resolved during design, development, or testing. However, some will remain undiscovered and will proceed to the live environment, and these may cause incidents. To manage errors that have arisen in the live environment, organizations have developed the problem management practice. The practice aims to identify and analyse errors in the organization's products in order to minimize their negative impacts on the services being provided.

2.2 TERMS AND CONCEPTS

Errors that may cause (or have already caused) incidents are called problems.

Definition: Problem

A cause, or potential cause, of one or more incidents.

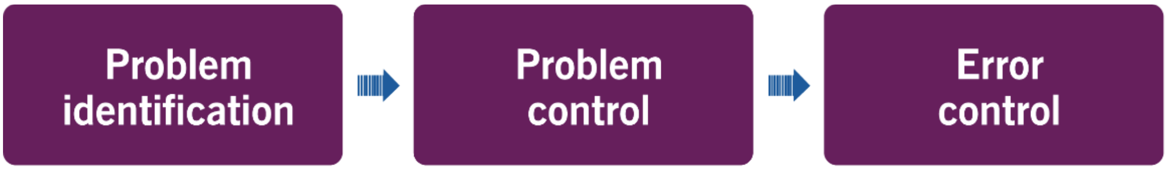

Figure 2.1 The three phases of the problem management practice

The problem management practice has three distinct phases, as shown in Figure 2.1.

2.2.1 Problem identification

There are two main approaches to problem identification. First, investigating the causes of incidents that have already happened. This approach starts with understanding the symptoms and then the causes. It aims to prevent incidents recurring, and may also contribute to the resolution of open incidents. This is known as reactive problem management because it is a reaction to incidents.

The second approach is to identify problems before they cause incidents, assess the related risks, and optimize the response with the aim of minimizing the probability and/or the impact of incidents. This is known as proactive problem management and is based on information about problems, specifically those that might be found in the live environment. The information sources may include:

- vendors informing about vulnerabilities in their products
- developers, designers, or testers discovering errors in live versions while working with the next versions
- user and specialist communities sharing their experiences of other organizations
- the monitoring of the infrastructure, discovering deviations in systems performance that do not yet qualify as incidents
- technical audits and other assessments.

Reactive or proactive?

The problem management practice is always reactive to problems: it does not prevent them from occurring the first time. The proactive/reactive distinction refers to how problem investigation relates to incidents:

- proactive problem management helps to prevent incidents from occurring the first time
- reactive problem management helps to prevent incidents from recurring and may help to resolve open incidents.

2.2.2 Problem control

Problem identification leads to the registration of a problem record. There can be a backlog of problems to analyse. Logged problems are accepted for analysis based on their initial categorization and prioritization. The initial categorization of a problem is likely to change after problem analysis, especially for problems that were registered based on incident (symptom) information.

Definitions

Task priority The importance of a task relative to other tasks. Tasks with a higher priority should be worked on first. Priority is defined in the context of all the tasks in a backlog.

Prioritization The action of selecting which tasks to work on first when it is impossible to assign resources to all tasks in the backlog.

Problem control focuses on the analysis of the problems. In reactive problem management, problem analysis uses information about the product architecture and configuration to identify configuration items (CIs) that are likely to cause the relevant incidents. The analysis is not limited to CIs and includes other factors, such as user behaviour, human errors, and procedure errors.

Proactive problem management usually starts with a better understanding of the CIs and other components of all four dimensions of service management which are suspected of causing incidents. For example, if a vendor informs the organization of a vulnerability in its software, problem control's task would be to identify how this software is used by the organization in order to assess the risks associated with the vulnerability and the potential impact on the services being provided.

When the problems have been analysed, they are assigned the status of ‘known error’.

Definition: Known error

A problem that has been analysed but has not been resolved.

Problem prioritization

Many problems have low urgency. This lack of urgency often keeps the problem backlog on hold; the service provider teams have more pressing tasks to do. However, it is important to ensure that identified problems are analysed and resolved. In the mid- and long-term perspectives, problems influence both the quality of the services being provided and the workload of the service provider.

There are a number of simple guidelines for problem prioritization:

- Evaluating the impact and urgency of a problem (and the time constraints for its investigation and resolution) is not prioritization. However, this evaluation is useful for prioritization and other important considerations, such as estimating the amount of time needed to perform the work.
- Prioritization is needed only when there is a resource conflict. Where there are sufficient resources to process every task within the time constraints, prioritization is unnecessary.
- Problems should be planned for processing using a single backlog, together with other tasks (planned and unplanned).
- Prioritization is a tool for assigning people to tasks in the context of a team. If a problem is handled by multiple teams, it will be prioritized within each team depending on resource availability, target completion time, and estimated processing time.
- Resource availability and estimated processing time are defined by the team. The target completion time depends on the problem’s impact; it may be defined when the problem is identified and initially categorized. The impact assessment and completion (resolution) time may change as problem analysis discovers new information.
- Visualization tools, such as Kanban, and Lean principles, such as the limiting of work in progress, are useful for effective prioritization.

These rules apply to all types of work (planned and unplanned) performed by the service provider’s specialist teams. It is important that they are agreed and followed by everyone involved in the organization’s service management activities, across all practices.

Problem analysis may find that errors have been removed from the organization’s environment or that they do not influence the services being provided. Following on from the above example, the organization may not use the vulnerable version of the software or the vulnerability may not affect the organization’s services. In these cases, the problem record may be closed after analysis. In other cases, it may remain open and error control may start.

Other important possible outputs of problem control are recommendations for incident resolution. Quite often, a better understanding of the causes of incidents helps to suggest a more effective resolution for those incidents, including workarounds.

Definition: Workaround

A solution that reduces or eliminates the impact of an incident or problem for which a full resolution is not yet available. Some workarounds reduce the likelihood of incidents.

Note that workarounds for incidents derived from problem analysis usually do not reduce the likelihood of incidents. Instead, they help to resolve incidents quicker and better when they occur. Workarounds that may help to prevent incidents are more likely to be identified at the error control stage.

2.2.3 Error control

When a problem has been analysed (i.e. the errors in the products have been localized and their impact on services has been assessed), it should be controlled. Problem records may be closed only if one of the following conditions is met:

- The problem is solved: the risk of incidents associated with the problem is removed or decreased to an acceptable level.
- The problem no longer affects the organization.

Note that although ‘known error’ is the state of a problem, some organizations prefer to have separate records for problem control and error control. In these cases, the problem record may be closed when problem analysis is complete, and the following activities may be registered in a related known error record. The above conditions for closure apply to known errors, regardless of whether they are problems.

Many known errors remain open for a long time if they cannot be efficiently resolved and they keep affecting services. In these cases, the organization may focus on maximizing the effectiveness and efficiency of incident handling (sometimes to the level of fully automated detection and resolution), but the problem records should remain open and be periodically reviewed.

The above approach to error control is valid where the costs of problem resolution may be higher than the costs of living with known errors and effective incident management. This is typical for problems associated with third-party components, especially where the third party is unresponsive or the components are unsupported. Conversely, where components are available for improvement and can be improved (especially software under the organization’s own control), known errors should be quickly removed.

Known errors are a part of an organization’s technical debt and should be removed where reasonably practicable.

Definition: Technical debt

The total rework backlog accumulated by choosing workarounds instead of system solutions that would take longer.

Error control ensures that the organization has sufficient up-to-date information about all the known errors in its products, including their statuses and their impacts on services. Error control optimizes problem resolution so that its costs and side-effects are balanced by its positive effects. Reviewing known errors periodically helps to identify changes in circumstances (such as business impacts, the availability of a permanent solution and the associated costs, and resource availability) that may trigger the re-assessment of the error and initiate its resolution.

The key outputs of error control are improvement initiatives and change requests, which initiate the resolution of problems. Some resolutions fix the errors in CIs and other product components. Others may introduce permanent workarounds: changes to the product configuration which do not fix the error but which reduce the likelihood of incidents to a minimum. The associated problem records may then be closed, but it is important to keep the knowledge about the errors available. This knowledge may be extremely valuable for future service design and when planning changes.

Permanent workarounds are normally used for components that the organization cannot fix (legacy systems, engineering infrastructure provided by third parties, etc.) but the use of permanent workarounds to prevent incidents increases an organization's technical debt and should be avoided wherever possible.

To summarize, possible types of problem mitigation are listed in Table 2.1.

Table 2.1 Approaches to problem mitigation

Mitigation approach	Applicability	Effect
Full permanent fix of the errors found. Problem record is closed.	Recommended approach for all CIs and other product components under the organization's full control. Highly recommended for software developed by the organization.	Incidents are prevented, side-effects are minimized, and quality of services is improved in the short-, mid-, and long-term perspectives.
Permanent workaround isolating the errors. Problem record may be closed or remain open.	May be recommended for CIs that cannot be fixed (third-party and/or legacy systems).	Incidents are prevented for the current product configuration; future designs and changes should consider the workarounds and may be limited by them.
Solutions are provided to optimize incident management. Problem record remains open.	Applicable to low-impact problems with very high costs for available permanent fixes or with no available fixes.	Incidents recur, but their impact is minimized. The known error should be periodically reviewed to ensure that recommended incident solutions are effective and there is still no permanent problem solution available.

2.2.4 Problem models

Different sources and types of problem may require different approaches to problem identification and control. For example, one or more of the following problem types may require a special approach to the problem management practice. These can be problems in:

- software
- hardware
- procedures
- third-party components
- consumer’s resources
- data
- data associated with sensitive information
- highly regulated services and systems.

To optimize the handling and resolution of these and other types of problems, a service provider often defines problem models. Problem models help to manage problems effectively and efficiently, often with better results because of the application of relevant proven and tested methods.

Definition: Problem model
A repeatable approach to the management of a particular type of problem.

The creation and use of problem models are important activities in the problem management practice. They are described in section 3.

2.3 SCOPE

The scope of the problem management practice includes:

- the identification and analysis of problems, including the analysis and control of known errors
- the initiation of changes to fix or reduce the impact of problems
- providing information about problems to the relevant stakeholders
- monitoring errors and the continual improvement of workarounds.

There are several activities and areas of responsibility that are not included in the problem management practice, although they are still closely related to problems. These are listed in Table 2.2, along with references to the practices in which they can be found. It is important to remember that ITIL practices are merely collections of tools to use in the context of value streams; they should be combined as necessary, depending on the situation.

Table 2.2 Activities related to the problem management practice described in other practice guides

Activity	Practice guide
Incident resolution	Incident management
Control and implementation of changes initiated to fix the problems	Change enablement Deployment management Infrastructure and platform management Release management Software development and management Other practices

Risk assessment and control	Risk management
Detection and control of errors in products before deployment to the live environment	Deployment management Service design Service validation and testing Software development and management
Communication of workarounds for incidents to users	Service desk

2.4 PRACTICE SUCCESS FACTORS

> Definition: Practice success factor
> A complex functional component of a practice that is required for the practice to fulfil its purpose.

A practice success factor (PSF) is more than a task or activity, as it includes components of all four dimensions of service management. The nature of the activities and resources of PSFs within a practice may differ, but together they ensure that the practice is effective.

The problem management practice includes the following PSFs:

- identifying and understanding the problems and their impact on services
- optimizing problem resolution and mitigation.

2.4.1 Identifying and understanding the problems and their impact on services

Organizations should understand the errors in their products because they may cause incidents and affect service quality and customer satisfaction. The problem management practice ensures problem identification and thus contributes to the continual improvement of products and services. This is more effective if performed proactively rather than reactively.

2.4.2 Optimizing problem resolution and mitigation

When problems have been identified, they should be handled effectively and efficiently. It is rarely possible to fix (remove) all the problems in an organization's products, but identification without resolution is significantly less valuable for the organization and its customers. A balanced approach should be defined for problem mitigation, namely one that considers the associated costs, risks, and impacts on the service quality.

2.5 KEY METRICS

The effectiveness and performance of the ITIL practices should be assessed within the context of the value streams to which each practice contributes. As with the performance of any tool, the practice's performance can only be assessed within the context of its application. However, tools can differ greatly in design and quality, and these differences define a tool's potential or capability to be effective when used according to its purpose. Further guidance on metrics, key performance indicators (KPIs), and other techniques that can help with this can be found in the measurement and reporting practice guide.

Key metrics for the problem management practice are mapped to its PSFs. They can be used as KPIs in the context of value streams to assess the contribution of the practice to the effectiveness and efficiency of those value streams. Some examples of key metrics are given in Table 2.3.

Table 2.3 Examples of key metrics for the practice success factors

Practice success factors	Key metrics
Identifying and understanding the problems and their impact on services	Number and impact of problems identified over the period Number and impact of incidents that are not associated with known errors Number and impact of incidents that require urgent problem investigation
Optimizing problem resolution and mitigation	Number and impact of incidents prevented by problem resolution Number and impact of incidents resolved with solutions provided by problem investigation Number and impact of known errors that remain open
Aggregated metric for the practice	Problem management productivity index[a]

[a] (N+C)/(O+C) – see the measurement and reporting practice guide for explanation and examples.

The correct aggregation of metrics into complex indicators will make it easier to use the data for the ongoing management of value streams, and for the periodic assessment and continual improvement of the problem management practice. There is no single best solution. Metrics will be based on the overall service strategy and priorities of an organization, as well as on the goals of the value streams to which the practice contributes.

3 Value streams and processes

3.1 VALUE STREAM CONTRIBUTION

Like any other ITIL management practice, the problem management practice contributes to multiple value streams. It is important to remember that a value stream is never formed from a single practice. The problem management practice combines with other practices to provide high-quality services to consumers. The main value chain activities to which the practice contributes are:

- deliver and support
- improve.

The contribution of the problem management practice to the service value chain is shown in Figure 3.1.

3.2 PROCESSES

Each practice may include one or more processes and activities that may be necessary to fulfil the purpose of that practice.

Definition: Process
A set of interrelated or interacting activities that transform inputs into outputs. A process takes one or more defined inputs and turns them into defined outputs. Processes define the sequence of actions and their dependencies.

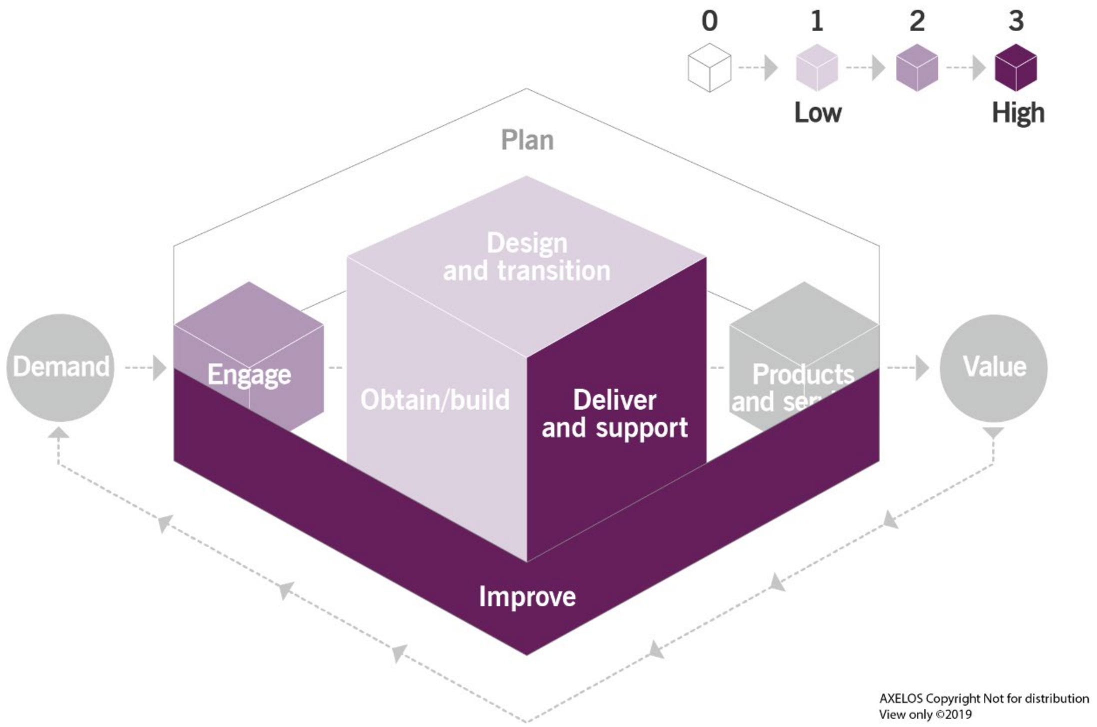

Figure 3.1 Heat map of the contribution of the problem management practice to value chain activities

Problem management activities form four processes:

- proactive problem identification
- reactive problem identification
- problem control
- error control.

3.2.1 Proactive problem identification

This process includes the activities listed in Table 3.1 and transforms the inputs into outputs.

Table 3.1 Inputs, activities, and outputs of the proactive problem identification process

Key inputs	Activities	Key outputs
Error information from vendors and suppliers Information about potential errors submitted by specialist teams Information about potential errors submitted by external user and professional communities Information about potential errors submitted by users Monitoring data Service configuration data	Review of the submitted information Problem registration Initial problem categorization and assignment	Problem records Feedback to the problem initiator

Figure 3.2 shows a workflow diagram of the process.

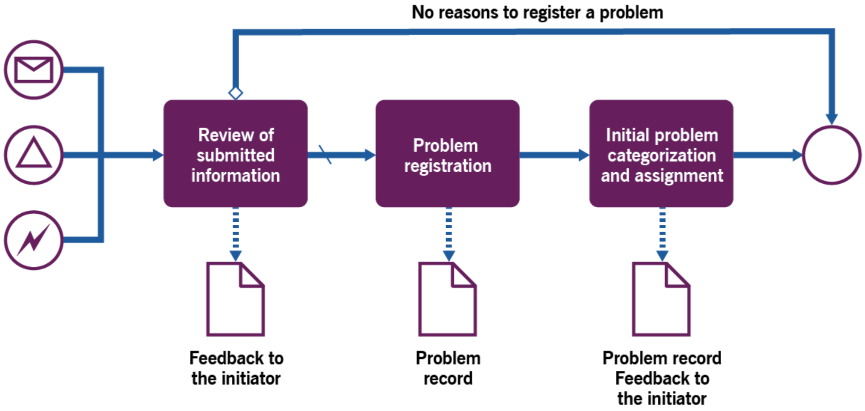
No reasons to register a problem
Review of submitted information
Problem registration
Initial problem categorization and assignment
Feedback to the initiator
Problem record
Problem record Feedback to the initiator

Figure 3.2 Workflow of the proactive problem identification process

Proactive problem identification is used to identify potential errors in the organization’s products based on sources other than incident records. Proactive problem identification and control can be considered and performed as a form of risk management which is focused on the vulnerabilities in the organization’s product: it includes the identification, assessment, and analysis of the vulnerabilities and the associated risks.

Possible sources of information about errors in an organization’s products are listed in Table 3.2.

Table 3.2 Sources of information for proactive problem identification

Source	Examples of information
Service designers, software developers, architects, and other teams working on the next versions of CIs and other components	Errors in the current live versions discovered during work on the subsequent versions Errors in the versions currently being deployed to the live environment that have been identified during testing but have not been fixed
Vendors of software and other CIs	Errors in the current live versions of the vendor’s systems and components
User and professional communities	Errors by other organizations using the same versions of systems and components
Monitoring data	Suspicious trends and deviations in the performance of services and CIs
Users	Vulnerabilities in the services being used

Where possible, proactive risk management activities should focus on the key systems and components with the highest potential impact on the organization and its customers. However, indications of errors in other systems should not be neglected. In complex technical environments designed for high availability, incidents may have multiple causes which are often the results of improbable combinations of improbable factors. Seemingly unimportant errors in non-core systems can contribute to serious incidents. Proactive problem identification should include the careful assessment of the probability and impact of the identified vulnerabilities.

Table 3.3 provides examples of the process activities.

Table 3.3 Activities of the proactive problem identification process

Activity	Example
Review of the submitted information	Depending on the source and the subject, the submitted information is reviewed by a specialist or a specialist group. The review includes checks for duplicates, applicability, common sense, and ongoing incidents potentially related to the submitted information. If the decision is made not to register a problem, the initiator may be notified (usually applicable in case of an active or ‘push’ submission; not applicable if the information was obtained or ‘pulled’ from external sources, such as vendor bulletins where nobody is expecting feedback).

Problem registration	If the need for problem control is confirmed, a problem record is registered. This can be done by a dedicated role or by a wider group of specialist roles.
Initial problem categorization and assignment	The person registering a problem performs the initial categorization. The information usually includes some of the following (if known or reasonably assumed): ● source ● description ● associated CIs and/or CI classes ● estimated impact and probability of incidents ● associated and potentially affected services ● impact on the organization and customers. Based on the initial categorization, the problem is assigned to a specialist group responsible for the associated CI, service, or product. Where applicable and expected, the problem initiator may be notified about the problem registration.

Who can register a problem?

There are several approaches to assigning responsibility for problem registration. One approach is to encourage every specialist to initiate and register problems. This would increase the number of improvements and improve the visibility of the errors in the organization’s products.

However, this may significantly increase the number of registered problems that nobody works with or that are incorrectly categorized. To prevent this, some organizations prefer to make one or more roles responsible for the initial filtering of the potential problems and for registering them. This approach may be effective as long as those in the roles have the resources they need, and can process information from various sources consistently and transparently.

Organizations can combine these approaches (and others) to balance the scope, throughput, and efficiency of problem identification.

3.2.2 Reactive problem identification

This process includes the activities listed in Table 3.4, and transforms the inputs into outputs.

Table 3.4 Inputs, activities, and outputs of the reactive problem identification process

Key inputs	Activities	Key outputs

Information about ongoing incidents Incident records and reports Monitoring data Service configuration data Service level agreements (SLAs)	Problem registration Initial problem categorization and assignment	Problem records

Figure 3.3 shows a workflow diagram of the process.

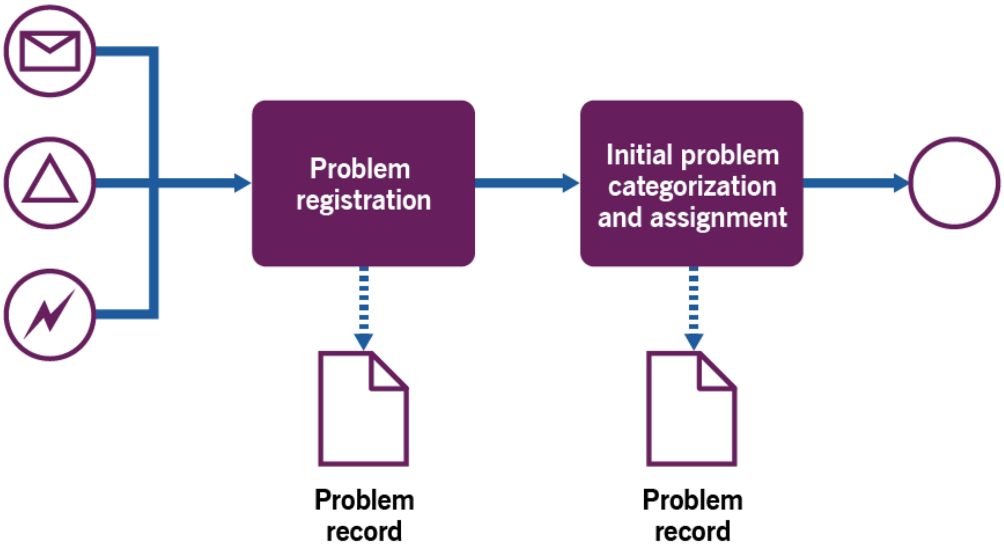

Figure 3.3 Workflow of the reactive problem identification process

Reactive problem identification uses information about past and ongoing incidents to investigate their causes. It can be triggered by an ongoing incident investigation that does not understand the nature of the incident; in this case, problem identification and control may be urgent. The incident management and problem management practices are used within a single value stream and are likely to involve the same (or overlapping) resources, including teams, tools, and procedures.

When based on the analysis of past incidents, problem identification may include statistical analysis, impact analysis, and trend analysis in various perspectives. This is to identify groups of incidents with possible common causes or other correlations.

The process varies slightly depending on the trigger. The variations are illustrated in Table 3.5.

Table 3.5 Activities of the reactive problem identification process

Activity	Triggered by ongoing incident	Triggered by incident records analysis
Problem registration	The team working on the incident identifies the need for problem investigation. In some cases, a problem record is linked to one or more incident records for tracking the investigation. It may be especially important where multiple incidents in numerous locations are being handled by different teams and require a coordinated problem investigation, or where problem investigation will be done by a dedicated team. In other cases, the team working with the incident may continue investigating the incident's causes and document the problem after the incident is resolved. The problem may still need to be registered, especially if the causes of the incident were not removed during incident resolution and new incidents may arise from the same problem.	A specialist team responsible for a system, service, or product performs regular reviews of the incident records associated with their area of responsibility. If they detect a reason for a problem investigation, they register a problem record. These reasons may include: ● a high number of similar incidents ● a high percentage of incidents resolved after the target resolution time ● major incidents ● availability below the target level.

Activity	Triggered by ongoing incident	Triggered by incident records analysis
Initial problem categorization and assignment	When registering a problem, the person doing so performs initial categorization. This usually includes some of the following (if known or reasonably assumed): ● description ● associated CIs and/or CI classes ● estimated impact and probability of incidents ● associated and potentially affected services ● impact on the organization and customers If the problem is registered before the problem investigation, the problem is assigned to the appropriate specialist group. If the problem is registered after the problem investigation, the information includes the steps made, the results, and the current status of the problem. If the problem is not solved at the time of registration, it is assigned to the appropriate group.	When registering a problem, the person doing so performs initial categorization. This usually includes some of the following (if known or reasonably assumed): ● description ● associated incidents and their solutions ● associated CIs and/or CI classes ● estimated impact and probability of future incidents ● associated and potentially affected services ● impact on the organization and customers ● estimated impact and probability of incidents Based on initial categorization, the problem is assigned to a specialist group, responsible for the associated CI, service, or product.

3.2.3 Problem control

This process focuses on the investigation of the problem. It includes the activities shown in Table 3.6 and transforms the inputs into outputs.

Table 3.6 Inputs, activities, and outputs of the problem control process

Key inputs	Activities	Key outputs
Problem records Service configuration data Technical information about CIs, products, and services Incident records Monitoring data	Problem investigation Known error communication	Problem records Known errors Incident solutions

Figure 3.4 shows a workflow diagram of the problem control process.

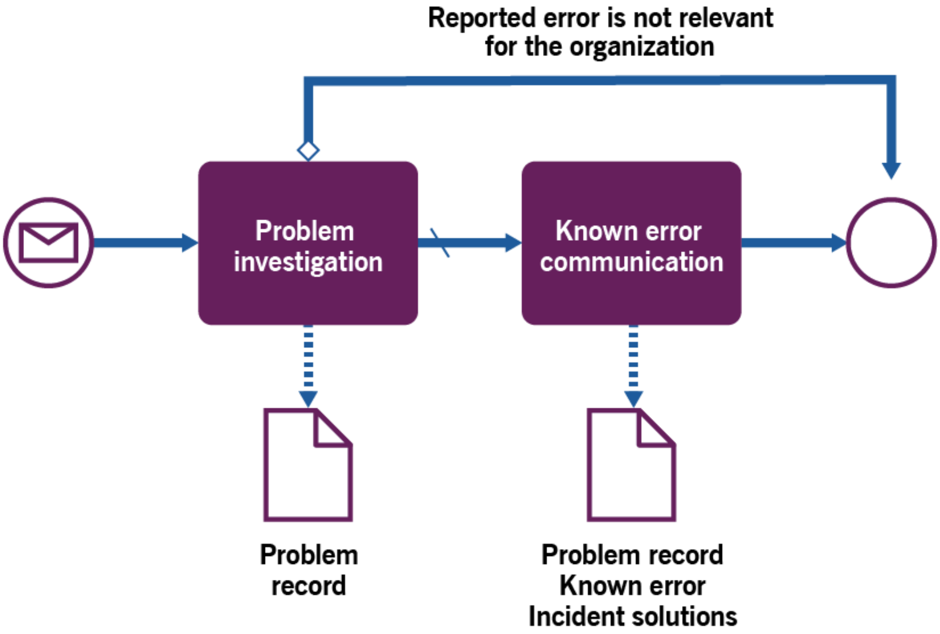

Figure 3.4 Workflow of the problem control process

Table 3.7 provides examples of the process activities.

Table 3.7 Activities of the problem control process

Activity	Example
Problem investigation	The specialist team assigned to the problem investigates the possible causes of the incident and/or verifies the reported errors in the CIs and the organization's other resources. The methods and procedures vary depending on how the problem has been identified. For problems identified reactively, localization starts with understanding which CIs may have errors causing past or ongoing incidents. For most problems identified proactively, CIs or CI classes would have been identified during their registration. After the problem is localized to the level of CIs, further diagnostics may be needed to identify errors within the suspicious CIs. This and the following activities may be performed by different teams (teams re-assigned based on the problem localization). If the reported problem is irrelevant to the organization (for example, a publicly reported vulnerability in software that does not affect the versions used by the organization), the problem record may be closed. If the investigated problem is relevant to the organization, it is assigned the known error status for further control and resolution. Actions and results of the investigations are recorded in the problem records.
Known error communication	The results of problem investigation are communicated to the problem initiator and relevant teams. These may include product development teams, technical specialists, the service desk team, users, and suppliers. If there are ongoing incidents associated with the problem that is being investigated, the results of the problem localization are communicated to the incident investigation teams. It is possible that understanding the errors is enough to define an incident resolution. In this case, a recommended solution for the incident should be registered in the problem records and communicated to the teams working with the incident.

To investigate problems, organizations use various analysis techniques. These may include:

- root cause analysis techniques, such as 5 Whys, Kepner and Fourie, and fault tree analysis
- impact analysis techniques, such as component failure impact analysis and business impact analysis
- risk analysis techniques.

It is important to remember that the concept of a single root cause has a very limited applicability in complex evolving environments. Quite often, incidents are caused by improbable combinations of improbable factors. Consequentially, the investigation of problems (especially in reactive problem management) should not be limited to the identification of the first possible cause of incidents. Problem investigation should always consider all four dimensions of service management.

Further guidance on the use of specific techniques for problem investigation can be found in supplementary ITIL publications.

3.2.4 Error control

This process focuses on the monitoring and control of the status of the known errors (problems that are analysed but not resolved) and their resolution. It helps to ensure that the negative impacts of the known errors on services are understood and minimized; the solutions for related incidents are effective; and the mitigation approach for the known error is valid, effective, and efficient.

This process includes the activities shown in Table 3.8 and transforms the inputs into outputs.

Table 3.8 Inputs, activities, and outputs of the error control process

Key inputs	Activities	Key outputs
Problem records Service configuration data Technical information about CIs, products, and services Incident records Monitoring data Knowledge management data	Problem solution development Problem resolution initiation Known error monitoring and review Problem closure	Problem records Problem models Change requests Improvement initiatives Problem solutions

Figure 3.5 shows a workflow diagram of the process.

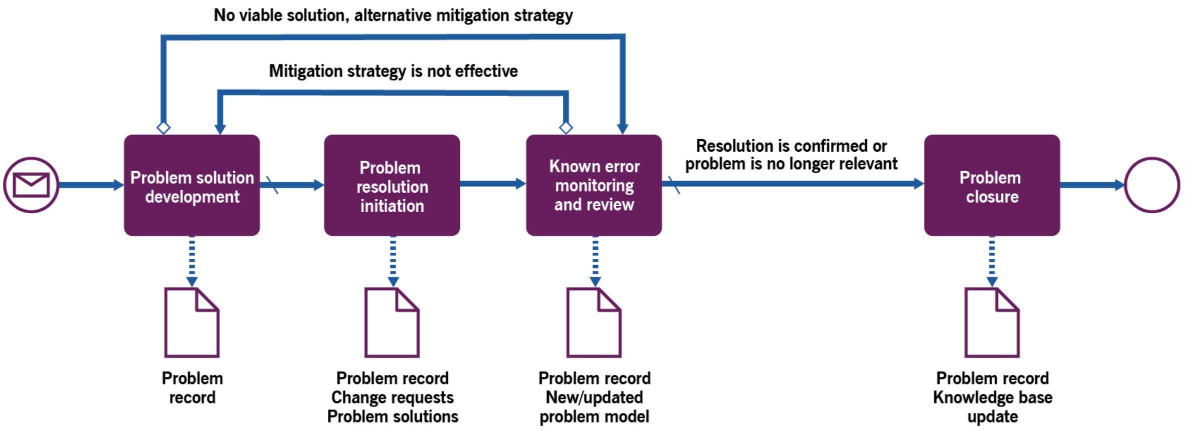
No viable solution, alternative mitigation strategy
Mitigation strategy is not effective
Problem solution development
Problem resolution initiation
Known error monitoring and review
Resolution is confirmed or problem is no longer relevant
Problem closure
Problem record
Problem record
Change requests
Problem solutions
Problem record
New/updated problem model
Problem record
Knowledge base update

Figure 3.5 Workflow of the error control process

Table 3.9 provides examples of the process activities.

Table 3.9 Activities of the error control process

Activity	Example
Problem solution development	The team (assigned or re-assigned based on the problem investigation) looks for a way to solve the problem. This includes defining an approach to the mitigation (see Table 2.1) and development of the actual solution within the selected approach. If there is no viable solution for the problem, the supporting information is recorded and the error goes to periodic review.
Problem resolution initiation	In most cases, problem resolution requires change. The responsible team submits change requests, following the organization's procedures for change initiation and implementation. In other cases, required actions are not classified as changes and can be initiated and performed following other procedures. Either way, the team initiates the actions required for the defined (and, if needed, approved) problem resolution. This initiation may need to be supported with relevant justification (including financial, risk, compliance, technical, and other considerations).

Known error monitoring and review

If a solution is approved for the known error

The implementation of the solution is controlled and confirmed using pre-agreed criteria. This is usually done by the team that initiated the resolution, or another pre-agreed role, such as problem manager.

For reactively identified problems, this can be done based on the change in incident dynamics (related incidents are resolved or prevented). For proactively identified problems, resolution control is based on the success of the initiated changes and may include a period of monitoring any service that might have been affected by the errors.

If the resolution of the problem is unconfirmed, the team returns to the problem solution development step of the process.

If no viable solution is found for the known error

An assigned specialist team should monitor the known error. This is usually the team responsible for the CI, service, or product with which the known error is associated. The team monitors the status of the known error as defined in the mitigation strategy. Monitored parameters may include:

- the dynamics of the associated incidents
- the effectiveness of the incident solutions
- the effectiveness of the problem workarounds
- changes in the statuses of the resources needed to solve a problem (budget, updates from the vendor, specialists, new infrastructure, etc.).

The team should conduct problem reviews periodically (in line with the agreed mitigation approach) or based on outstanding monitoring results.

If the review confirms that the mitigation approach is valid and up to date (the problem exists, the impact assessment is up to date, incident solutions are effective, the problem workaround is effective, and no viable problem fix is available), then known error monitoring continues.

If the mitigation approach becomes invalid, the problem solution development activity is initiated to review and redefine the mitigation approach. This may include developing and implementing a problem solution or updating the incident solutions for any associated incidents.

If the problem no longer exists (for example, it has been removed with planned software or hardware updates or by decommissioning the affected CIs), problem closure is initiated.

If the problem demonstrated a new pattern that suggests the amendment or creation of a problem model, a problem model is documented and communicated as part of the problem review activity.

Problem records are updated with monitoring data.

Activity	Example
Problem closure	The team (or specialist) responsible for the problem documents the problem review results and formally closes the problem record. If the resolution is confirmed, the team documents the resolution control results and formally closes the problem record. Closed problem records should be available as part of the organization's knowledge base, especially if there is a chance that similar problems may recur.

Problem management activities are performed by the service provider, as described in Tables 3.3, 3.5, 3.7, and 3.9. They may involve suppliers and partners, and sometimes customers and users. These activities are also supported (and sometimes fully or largely automated) by tools and technologies. All are described in the following sections.

4 Organizations and people

4.1 ROLES, COMPETENCIES, AND RESPONSIBILITIES

The practice guides do not describe the practice management roles such as practice owner, practice lead, or practice coach. They focus instead on the specialist roles that are specific to each practice. The structure and naming of each role may differ from organization to organization, so any roles defined in ITIL should not be treated as mandatory, or even recommended. Remember, roles are not job titles. One person can take on multiple roles and one role can be assigned to multiple people.

Roles are described in the context of processes and activities. Each role is characterized with a competency profile based on the model shown in Table 4.1.

Table 4.1 Competency codes and profiles

Competency code	Competency profile (activities and skills)
L	Leader Decision-making, delegating, overseeing other activities, providing incentives and motivation, and evaluating outcomes
A	Administrator Assigning and prioritizing tasks, record-keeping, ongoing reporting, and initiating basic improvements
C	Coordinator/communicator Coordinating multiple parties, maintaining communication between stakeholders, and running awareness campaigns
M	Methods and techniques expert Designing and implementing work techniques, documenting procedures, consulting on processes, work analysis, and continual improvement
T	Technical expert Providing technical (IT) expertise and conducting expertise-based assignments

Two practice-specific roles may be found in organizations: problem manager and problem coordinator. These roles are often introduced in organizations where the number of problems is high. In other organizations, problem management activities are coordinated by a person or a team responsible for the CIs, service, or product with which the problem is associated; this may be the resource owner, service owner, or product owner respectively.

4.1.1 Problem manager role

Where a dedicated problem manager role is defined, it is usually assigned to specialists combining good knowledge of the organization’s products (architecture, configurations, and interdependencies) with solid analytical skills (the ability and authority to coordinate teamwork and provide good risk management). The competency profile for this role is TMAC. This role is usually responsible for managing and coordinating the specialist activities in the problem management processes, including:

- conducting and coordinating problem registration based on the submitted information
- the initial categorization of the problems
- coordinating problem investigation and solution implementation control

- coordinating the communication with the teams responsible for incident resolution and change implementation
- developing and communicating problem models, where applicable
- coordinating known error monitoring and review
- the formal problem closure.

4.1.2 Problem coordinator role

In more complex organizations, some responsibilities for the problem management practice may be delegated to the problem coordinator. The problem coordinator focuses on routine problem management activities, such as the review of submitted information about possible problems, problem review, and problem closure.

Examples of other roles which can be involved in the problem management activities are listed in Table 4.2, together with the associated competency profiles and specific skills.

Table 4.2 Examples of roles with responsibility for problem management activities

Activity	Responsible roles	Competency profile	Specific skills
Proactive problem identification process			
Review of the submitted information	CI owner Problem coordinator Problem manager Product owner Service owner	T	Good knowledge of the product, including its architecture and configuration
Problem registration	CI owner Problem coordinator Problem manager Product owner Service owner	TA	Knowledge of the registration tools and procedures
Initial problem categorization and assignment	CI owner Problem coordinator Problem manager Product owner Service owner	TAC	Good knowledge of the product, service architecture, and business impact Understanding of the responsibilities and competencies across the teams
Reactive problem identification process			
Problem registration	CI owner Incident manager Problem coordinator Problem manager Product owner Service owner	TA	Knowledge of the registration tools and procedures

Activity	Responsible roles	Competency profile	Specific skills
Initial problem categorization and assignment	CI owner Incident manager Problem coordinator Problem manager Product owner Service owner	TAC	Good knowledge of the product, service architecture, and business impact Understanding of the responsibilities and competencies across the teams
Problem control process			
Problem investigation Problem coordinator Problem manager Product owner Service owner Supplier Technical specialist	CI owner	CT	Good knowledge of the product, service architecture, and business impact Good knowledge of diagnostic, investigation, and analysis methods and tools
Known error communication	CI owner Incident manager Problem coordinator Problem manager	TC	Understanding of stakeholders and responsibilities Knowledge of the communication tools and procedures
Error control process			
Problem solution development	CI owner Problem coordinator Problem manager Product owner Service owner Supplier Technical specialist	TMC	Good knowledge of the product and service architecture, configuration, and technical details Creativity Systems thinking
Problem resolution initiation	CI owner Problem coordinator Problem manager Product owner Service owner Technical specialist	CT	No specific skills required
Known error monitoring and review	CI owner Problem coordinator Problem manager Product owner Service owner Supplier Technical specialist	TAC	Good knowledge of the product and service architecture and business impact
Problem closure	CI owner Problem coordinator Problem manager Product owner Service owner	TCA	Good knowledge of the product, service architecture, and business impact

4.2 ORGANIZATIONAL STRUCTURES AND TEAMS

It is unusual to see a dedicated organizational structure for the problem management practice, although the role of problem manager is sometimes associated with a formal job title. This is typical for organizations with complex bureaucracy and a significant number of problems to manage. Many organizations find it useful to form temporary teams to investigate high-impact problems and/or to develop solutions.

In product-focused organizations, problem management job titles and roles are not typically adopted. Instead, this practice is integrated in the day-to-day activities of the product development and management teams. It is automated wherever possible.

5 Information and technology

5.1 INFORMATION EXCHANGE

The effectiveness of the problem management practice is based on the quality of the information used. This information includes, but is not limited to, information about:

- products and services and their architecture and design, including configuration information
- customers and users
- partners and suppliers, including contract and SLA information on the services they provide
- ongoing and past incidents
- planned, ongoing, and past changes
- a third-party's products and components, including vulnerabilities and incidents.

This information may take various forms. The key inputs and outputs of the practice are listed in section 3.

5.2 AUTOMATION AND TOOLING

In most cases, the problem management practice can significantly benefit from automation. Where this is possible and effective, it may involve the solutions outlined in Table 5.1.

Table 5.1 Automation solutions for problem management activities

Process activity	Means of automation	Key functionality	Impact on the effectiveness of the practice
Proactive problem identification process			
Review of the submitted information	Monitoring and event management tools, user portals and other user interfaces, workflow management and collaboration tools	Collection and overview of information from various sources, including data analysis and team collaboration	High
Problem registration	Workflow management and collaboration tools	Management of problem records integrated with other service management data	High
Initial problem categorization and assignment	Workflow management and collaboration tools, and configuration management tools	Management of problem records integrated with other service management data, backlog management, communication, and collaboration support	High
Reactive problem identification process			

Process activity	Means of automation	Key functionality	Impact on the effectiveness of the practice
Problem registration	Workflow management and collaboration tools	Machine-learning-based problem identification based on analysis of past and ongoing incidents Management of problem records integrated with other service management data	High
Initial problem categorization and assignment	Workflow management and collaboration tools, and configuration management tools	Management of problem records integrated with other service management data, backlog management, communication, collaboration support, and CI impact assessment	High
Problem control process			
Problem investigation	Diagnostic and analytical tools, and configuration management tools	Dependencies analysis, what-if analysis, cause-and-effect analysis, and modelling	High
Known error communication	Workflow management and collaboration tools	Communication and collaboration support	Medium
Error control process			
Problem solution development	Diagnostic and analytical tools, configuration management tools, and design tools	Solution design and validation	Medium to very high, depending on the solution architecture
Problem resolution initiation	Workflow management and collaboration tools	Communication and collaboration support	Medium

Process activity	Means of automation	Key functionality	Impact on the effectiveness of the practice
Known error monitoring and review	Monitoring and event management tools, workflow management and collaboration tools, and automated testing tools	Collection and overview of information from various sources, data analysis, and team collaboration Verification that known errors exist and workarounds work	Medium to high
Problem closure	Workflow management and collaboration tools	Communication and collaboration support, automatic posts into collaboration tools	Medium

6 Partners and suppliers

Very few services are delivered using only an organization's own resources. Most, if not all, depend on other services, often provided by third parties outside the organization (see section 2.4 of ITIL Foundation: ITIL 4 Edition for a model of a service relationship). Relationships and dependencies introduced by supporting services are described in the ITIL practices for service design, architecture management, and supplier management. Information about dependencies on third-party services is used in all problem management processes.

The problem management practice often discovers errors in third-party products used by the organization. The possibility of solving those errors, and the effectiveness of the solution, depends on multiple factors, including:

- the architecture of the solution
- the flexibility of the supplier
- the importance of the service relationship with the organization for the supplier
- the contract terms and conditions.

It is important to understand how the organization depends on third-party components and how it aims to establish effective and efficient collaboration with its key suppliers and partners around many activities, including those of the problem management practice.

Problem models should define how third parties are involved in problem control and how the organization ensures effective collaboration. This depends on the architecture and design solutions for products, services, and value streams. Quite often, after the correct model is selected for a problem, further consideration of third-party dependencies is needed within the processes of problem and error control.

Where organizations aim to ensure fast and effective problem management, they usually try to agree close cooperation with their partners and suppliers, removing formal bureaucratic barriers in communication, collaboration, and decision-making (see the supplier management practice guide for more information).

7 Important reminder

Most of the content of the practice guides should be taken as a suggestion of areas that an organization might consider when establishing and nurturing their own practices. The practice guides are catalogues of topics that organizations might think about, not a list of answers. When using the content of the practice guides, organizations should always follow the ITIL guiding principles:

- focus on value
- start where you are
- progress iteratively with feedback
- collaborate and promote visibility
- think and work holistically
- keep it simple and practical
- optimize and automate.

More information on the guiding principles and their application can be found in section 4.3 of ITIL Foundation: ITIL 4 Edition.

8 Acknowledgements

AXELOS Ltd is grateful to everyone who has contributed to the development of this guidance. These practice guides incorporate an unprecedented level of enthusiasm and feedback from across the ITIL community. In particular, AXELOS would like to thank the following people.

8.1 AUTHORS

Barry Corless, Roman Jouravlev, Andrew Vermes.

8.2 REVIEWERS

James Ainsworth, Akshay Anand, Sofi Fahlberg, Michael G. Hall, Steve Harrop, Piia Karvonen, Anton Lykov, Paula Määttänen, Caspar Miller, Christian F. Nissen, Mark O'Loughlin, Tatiana Orlova, Elina Pirjanti, Stuart Rance.